Spectral Interpretations of Subgraphs for Threat Discoveries (finding structure in graphs)

Briefing at NSF Gathering

Algorithms for Threat Detections (ATD)

Sept. 15th, 2017



Prof. Peter Chin at Boston University, CS Dept.

Prof. Van Vu at Yale Univ., Math Dept.

Prof. S.-T. Yau at Harvard University, Math Dept.

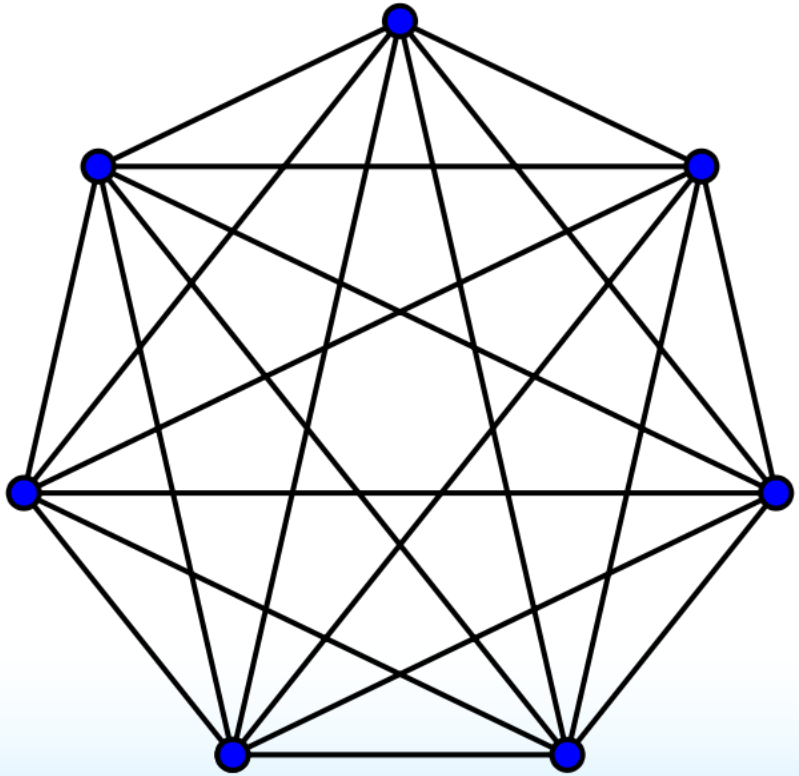
Dr. Mark Kempton at Harvard University, Center of Mathematics & Scientific Applications

Mr. Daniel Montealegre at Yale Univ., Math Dept.

Mr. Gavin Brown at Boston University, CS Dept.

Let's start counting….

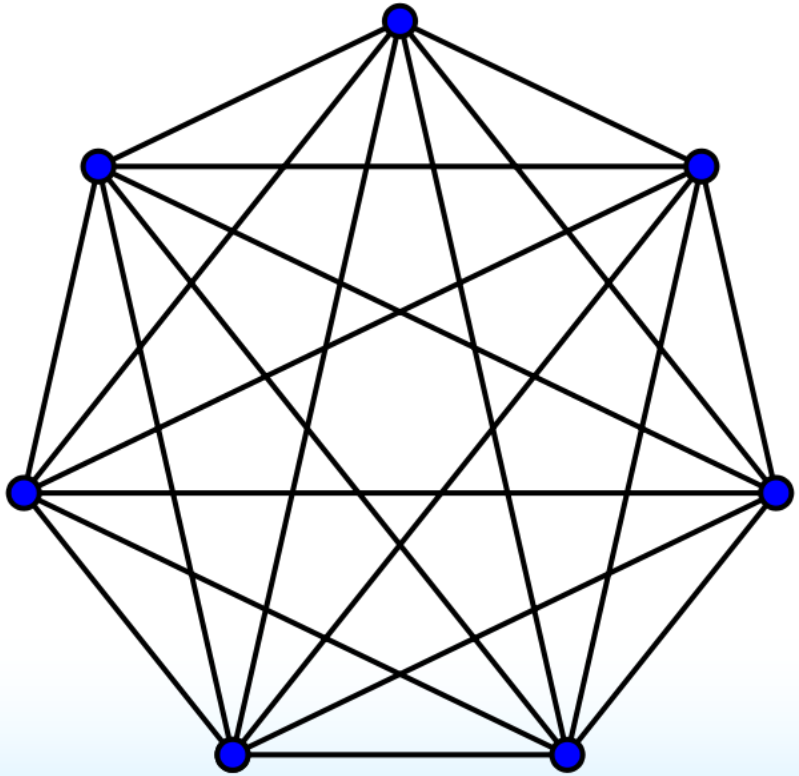
Q: What's the maximum number of edges in an n-vertex simple graphs?



Let's start counting….

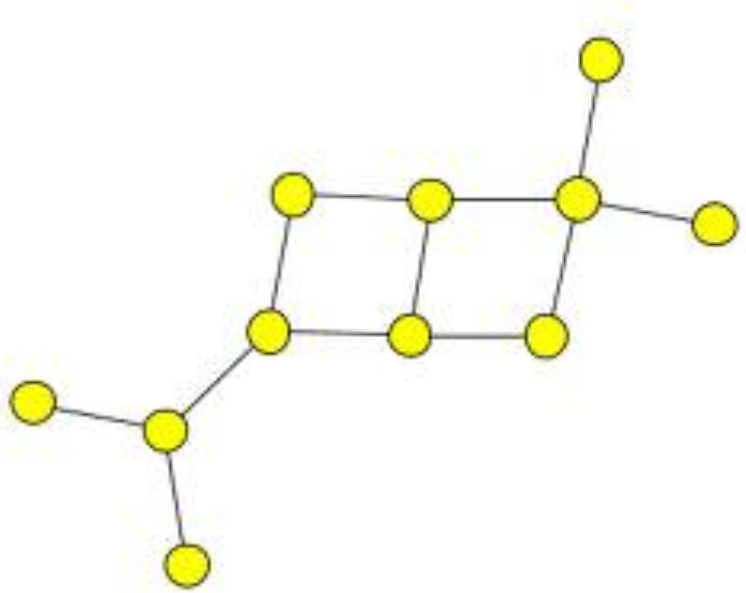
Q: What's the maximum number of edges in an n-vertex simple graphs?

A: *n(n-1)/2*



Let's start counting more….

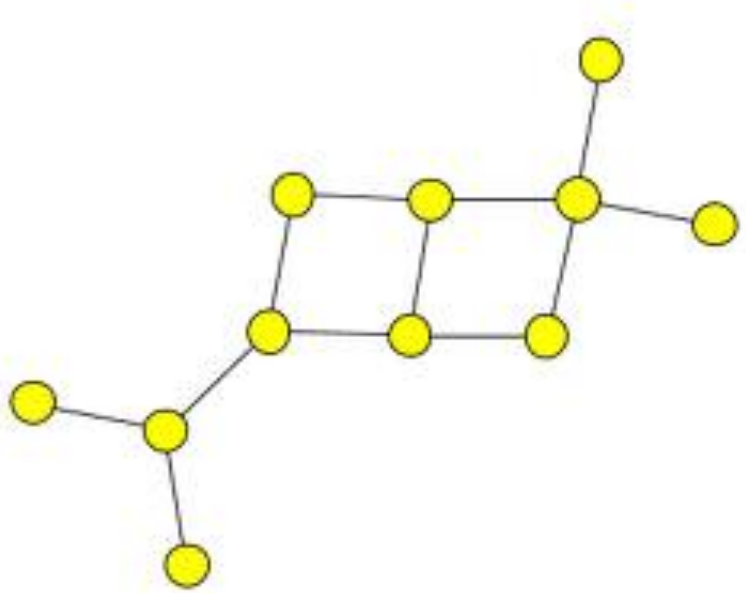
Q: What's the maximum number of edges in an n-vertex simple graphs that are triangle free?



Let's start counting more….

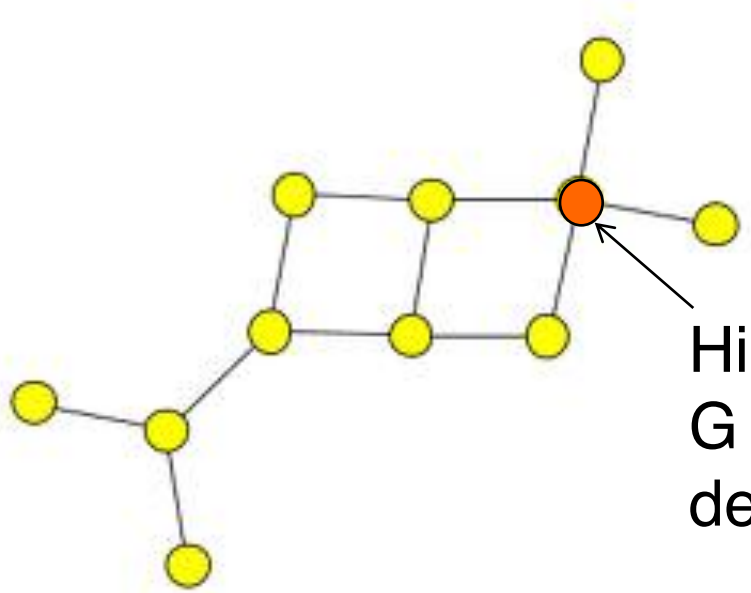
Q: What's the maximum number of edges in an n-vertex simple graphs that are triangle free?

Hint #1: G has 11 vertices!



Let's start counting more….

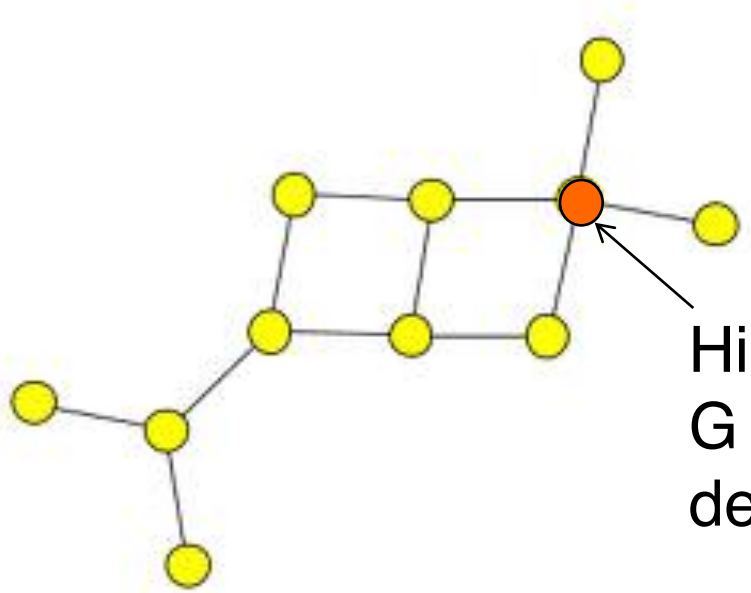
Q: What's the maximum number of edges in an n-vertex simple graphs that are triangle free?



Hint #2:
G has max degree of 4

Let's start counting more….

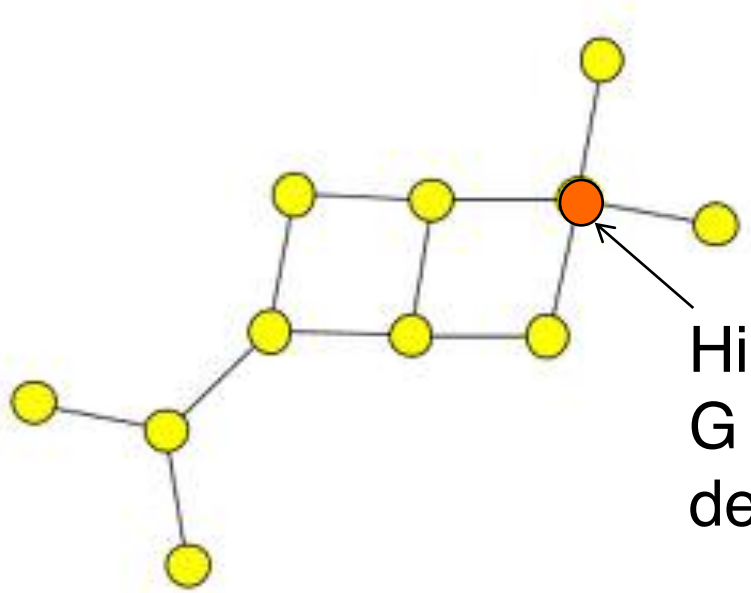
Q: What's the maximum number of edges in an n-vertex simple graphs that are triangle free?



Hint #2:
G has max degree of 4

Let's start counting more….

Q: What's the maximum number of edges in an n-vertex simple graphs that are triangle free?



Hint #2:
G has max degree of 4

A: $[11^2/4] = 31$

Let's start counting more….

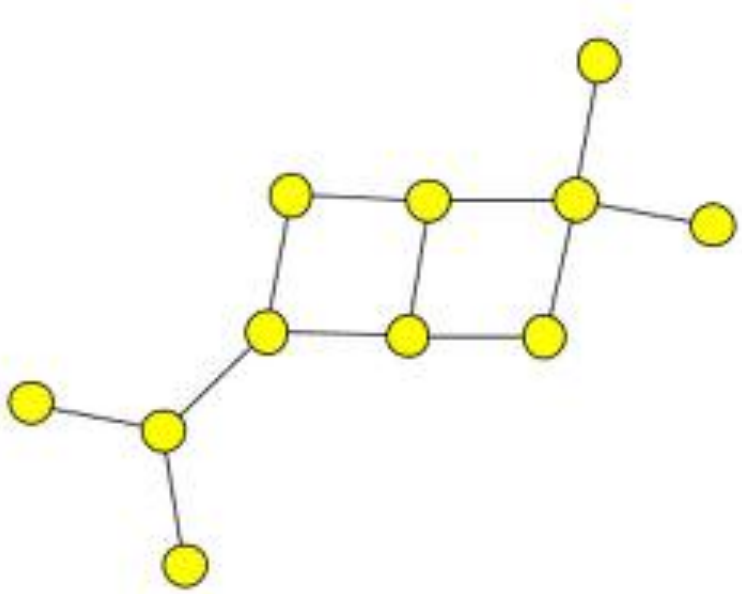
Q: What's the maximum number of edges in an n-vertex simple graphs that are triangle free? Mantel (1904)

Proof of Theorem 1: Let G be a triangle-free graph. Let x be a vertex of G with maximum degree. Let $k = d(x)$ be the degree of x. Let $N(x)$ be the set of neighbors of x. Since G is triangle-free, there are no edges whose endpoints are both in $N(x)$. $\overline{N(x)}$ forms a vertex cover. The number of edges $e(G)$ satisfies

$$\begin{aligned} e(G) &\leq \sum_{y \in N(x)} d(y) \\ &\leq \sum_{y \in N(x)} k \\ &= k \cdot |\overline{N(x)}| \\ &= k(n-k) \\ &\leq \left(\frac{k+(n-k)}{2}\right)^2 \\ &= \frac{n^2}{4} \end{aligned}$$

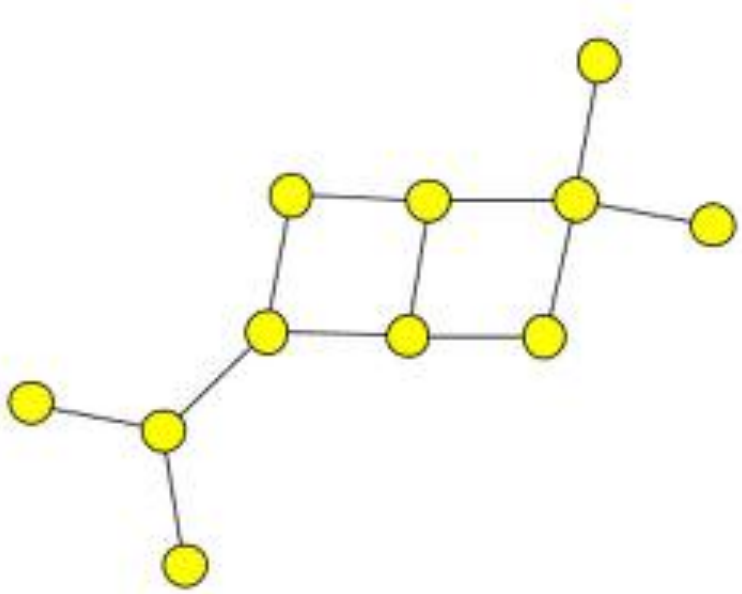
Let's keep going….

Q: How many triangles are there in this graph?



Let's keep going….

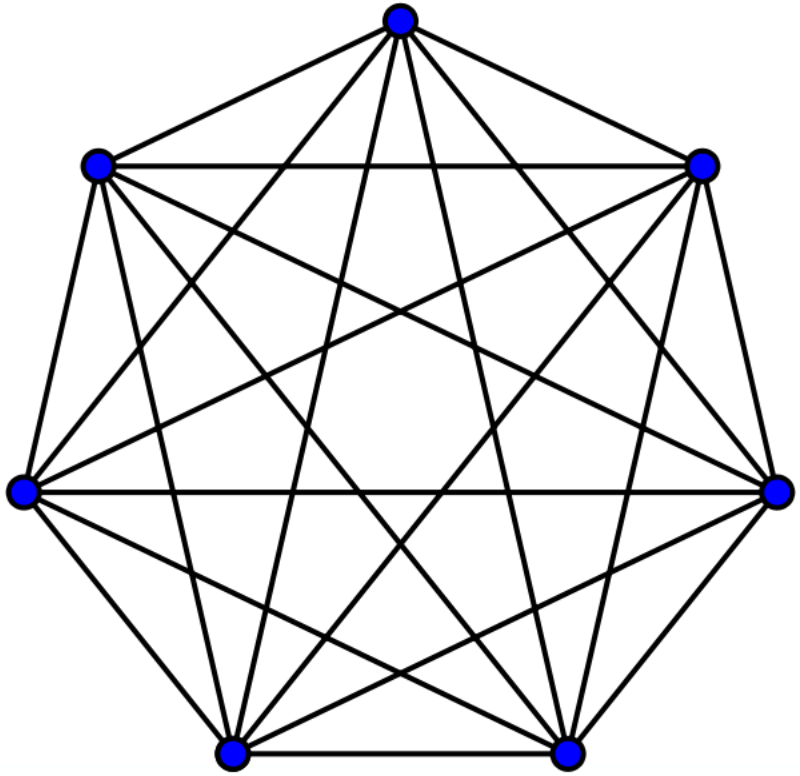
Q: How many triangles are there in this graph?



A: 0

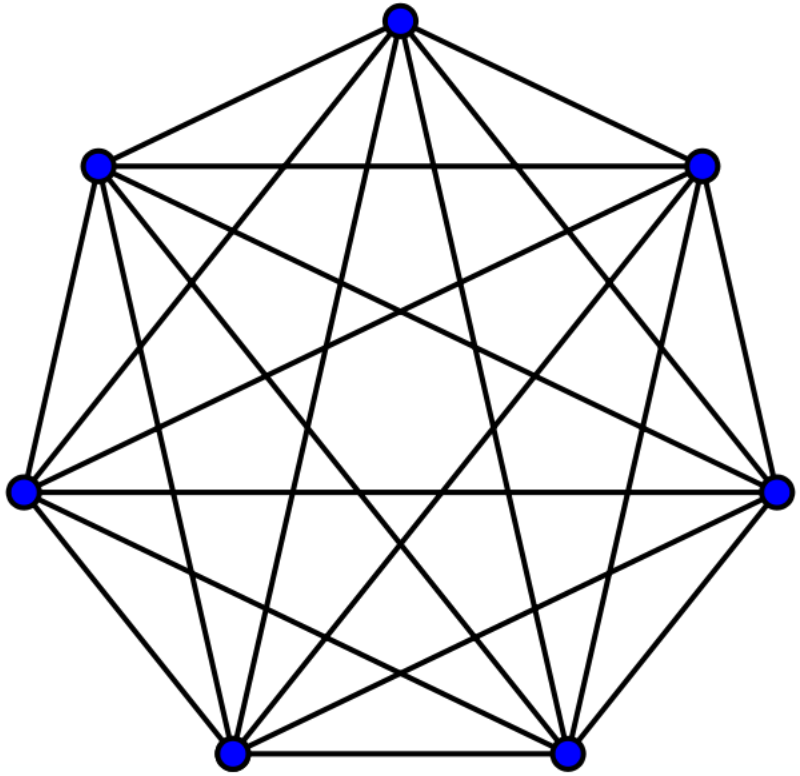
Let’s keep going….

Q: How many triangles are there in this graph?



Let's keep going….

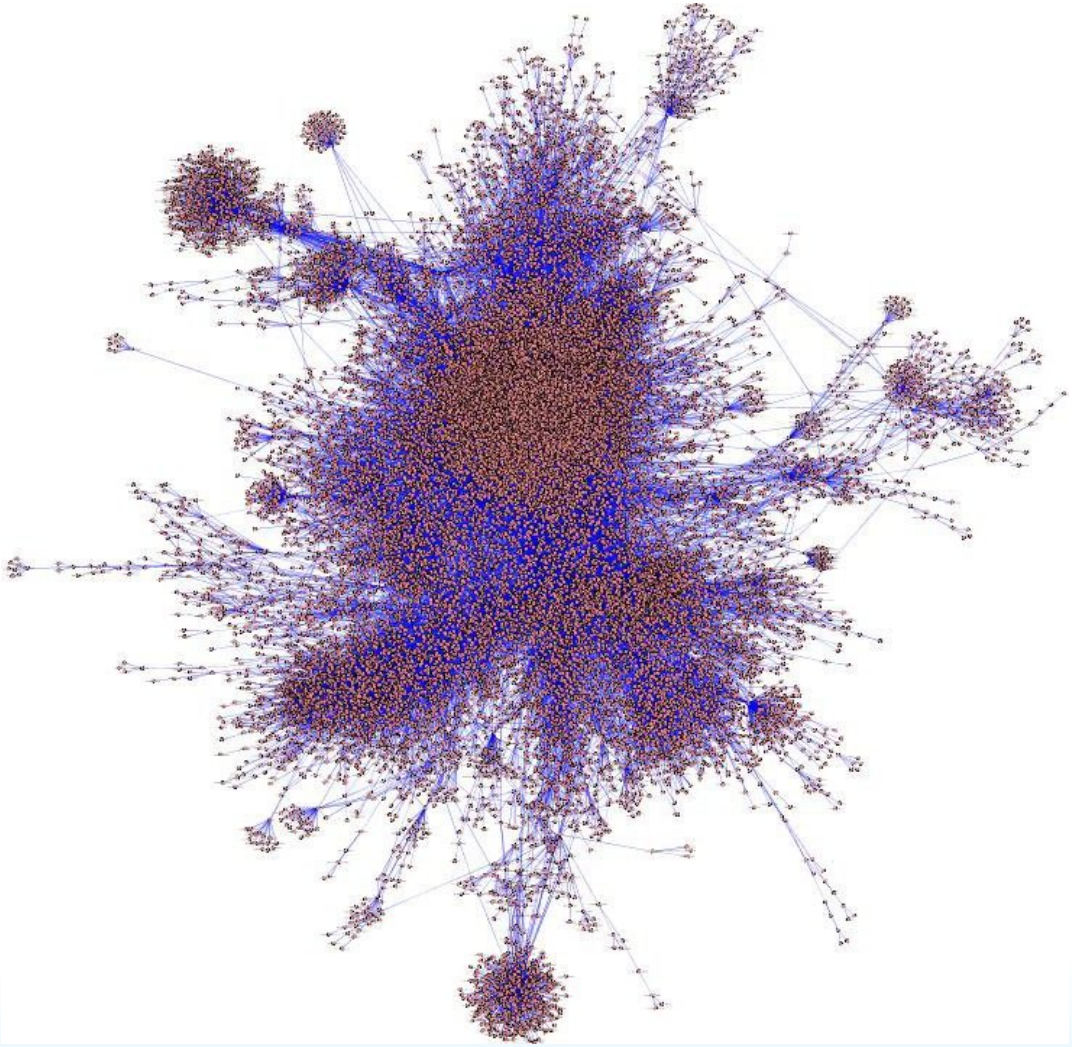
Q: How many triangles are there in this graph?



A: 7!/3!4! = 35

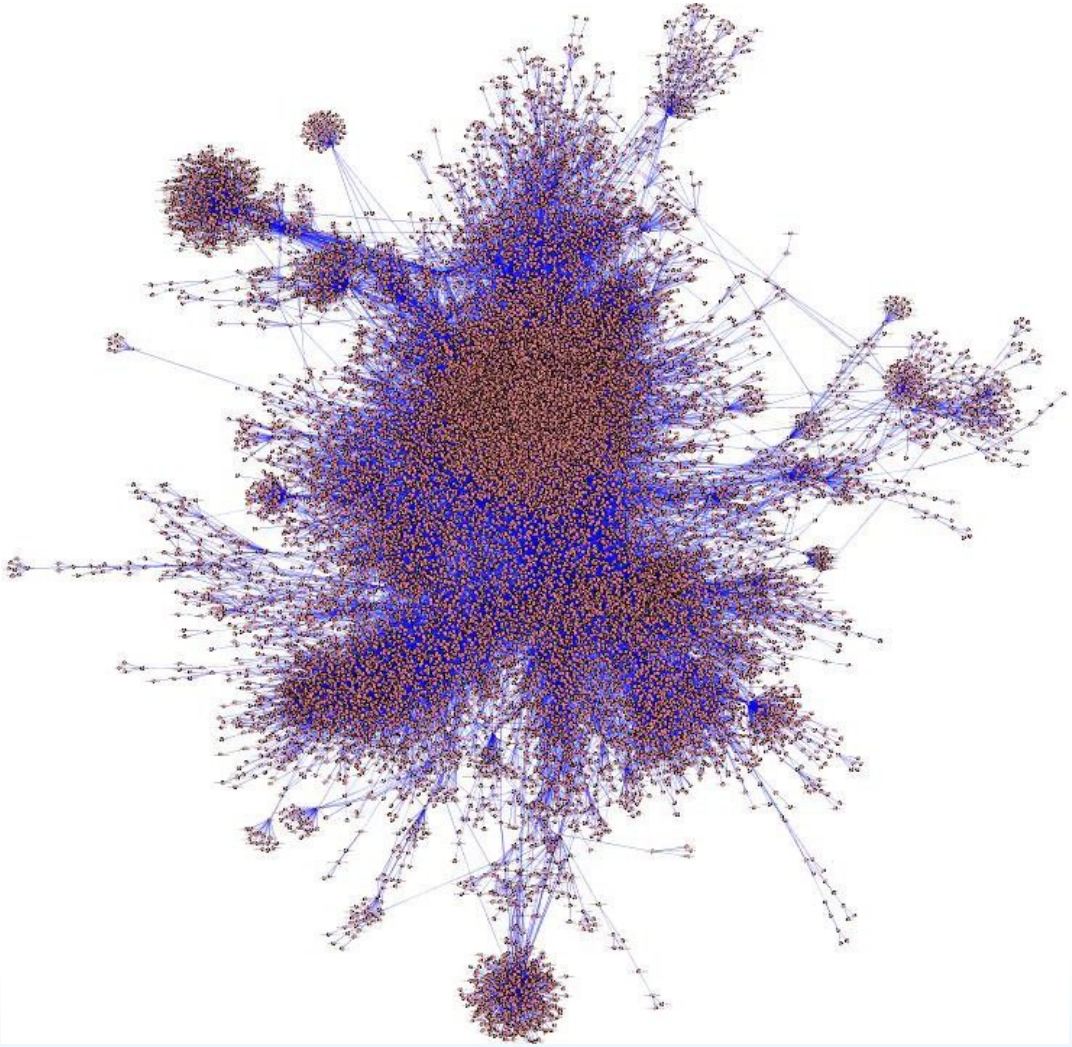
Let's keep going….

Q: How many triangles are there in this graph?



Let's keep going….

Q: How many triangles are there in this graph?



A: Good luck!

Well.. It's not that hopeless….

Things people tried:

- Simply enumerate C(n,3) edges of G

Well.. It's not that hopeless….

Things people tried:

- Simply enumerate C(n,3) edges of G – not a good idea $O(n^3)$

Well.. It's not that hopeless….

Things people tried:

- Simply enumerate C(n,3) edges of G
- NODEITERATOR
  - computes for each node its neighborhood and then sees how many edges exist among its neighbors. This algorithm runs asymptotically in $O(d^2_{max}n)$, where d_{max} is the maximum degree in G

Well.. It's not that hopeless….

Things people tried:

- Simply enumerate C(n,3) edges of G
- NODEITERATOR
  - computes for each node its neighborhood and then sees how many edges exist among its neighbors. This P algorithm runs asymptotically in $O(d^2_{max}n)$, where d_{max} is the maximum degree in G.
- EDGEITERATOR
  - checks each edge (u, v) and computes the common neighbors of the nodes u and v.
  - Asymptotically runs in the the same time with the NODEITERATOR but can be improved to $O(m^{3/2})$ using a simple hashing argument.

Well.. It's not that hopeless….

Things people tried:

-Fast matrix multiplication

- Alon et al. gave an algorithm with $O(m^{2p/(p+1)})$ where the best known p is 2.37, the exponent of the state-of the-art algorithm for matrix multiplication…

-Streaming algorithms

- perform one or at most a constant number of passes over the graph stream (e.g. edges arriving one at a time $\mathbf{\{e_1, e_2, .., e_m\}}$) [Yossef et al.]

-Semi-streaming algorithms

- Bechetti et al

Unfortunately….

- Graph algorithms that calculate basic graph properties are not often scalable for graphs containing billions of nodes.
- A sound mathematical theory is still desirable for analysis of graph dynamics.
- Several current approximation algorithms for NP problems do not have known accuracy.
- And more…

- What Next?

Enter the man….

- Introducing 2012 winner of Abel Prize….
- Drum rolls please….

Our first approach – Approximate the graph

- Introducing 2012 winner of Abel Prize….
- Drum rolls please….

Endre Szemeredi

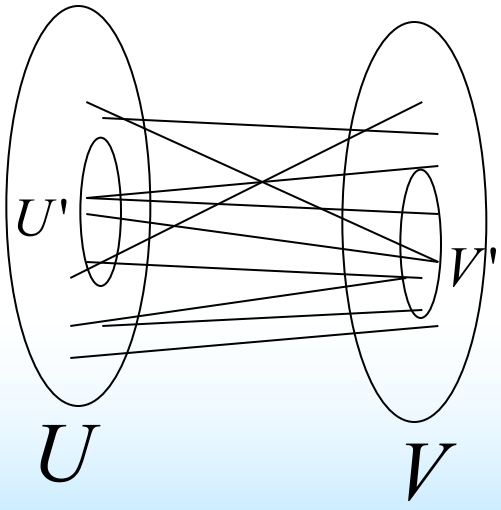
The density of a bipartite graph $G(V,U,E)$ is

$$d(U,V)=\frac{E(U,V)}{|U|\cdot|V|}$$

We say it is ε - regular if for every

$U'\subseteq U, V'\subseteq V$ with $|U'|>\varepsilon|U|, |V'|>\varepsilon|V|$

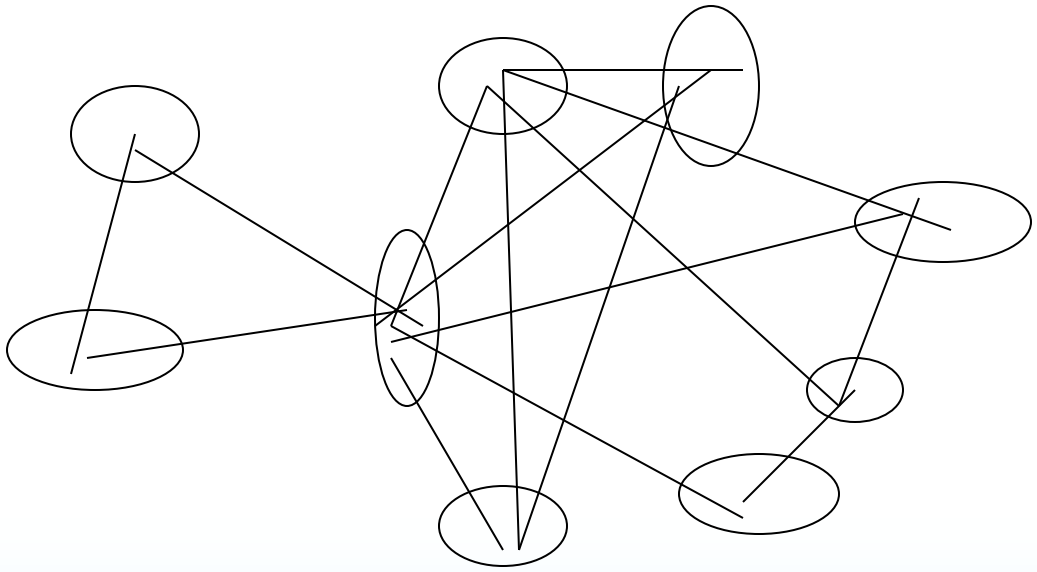
$$\left|d(U,V)-d(U',V')\right|<\varepsilon$$



Regular Partitions

A multi-partite graph on vertex sets $V_1, \cdots, V_k$

is ε -regular if all but $\varepsilon\binom{k}{2}$ of the pairs are regular



Szemerédi's Regularity Lemma:

$\forall\, k, \varepsilon\ \exists\, \mathrm{N}_0, \mathrm{K}$, such that for every graph $G = (V, E)$ with $|V| > N_0$

V can be partitioned into t almost equal parts with $k < t < K$ such that the resulting induced multipartite graph is ε - regular.

Szemerédi's Regularity Lemma:

$\forall\, k, \varepsilon\ \exists\, \mathrm{N}_0, \mathrm{K}$, such that for every graph $G = (V, E)$ with $|V| > N_0$

V can be partitioned into t almost equal parts with $k < t < K$ such that the resulting induced multipartite graph is ε - regular.

Q: How do we get this Partition?

Szemerédi's Regularity Lemma:

$\forall\, k, \varepsilon\, \exists\, \mathrm{N}_0, \mathrm{K}$, such that for every graph $G = (V, E)$ with $|V| > N_0$ V can be partitioned into t almost equal parts with $k < t < K$ such that the resulting induced multipartite graph is ε - regular.

Fundamental results such as of Szemeredi's regularity lemma lack computationally efficient algorithms that limit its usefulness of real-world applications.

New Insights Using Spectral Theory…

-Significant eigenvalues, which capture the essence of large graphs, live outside the essential spectral radius, number far less than the size of graphs, and there is an identifiable separation between significant and insignificant eigenvalues.

-Using the significant eigenvalues of a dense graph's adjacency matrix, the partition described in Szemeredi's regularity lemma may be quickly approximated to capture the essential information of a graph.

So what…

-We may be able to provide computationally efficient and scalable methods to approximate the partitions of Szemeredi's regularity lemma to identify a graph's essential information.

-We will efficiently compute various graph properties and apply known graph theory to the scalable partition thus obtained.

Our approach in a nutshell

- Consider a family of graphs $\{G_n\}$, where G_n is a graph on n vertices.
- List all eigenvalues of the adjacency matrix of G_n such that the absolute values are in decreasing order:

$$|\lambda_1| \geq |\lambda_2| \geq \cdots \geq |\lambda_n|.$$

- An increasing function $f(n)$ is a coarse spectral bound of G_n if

$$|\{i : |\lambda_i(G_n)| \geq f(n)\}| = o(n).$$

Some properties of Coarse Spectral Radius

-Monotone: If **f**(**n**) $\succeq$ **g**(**n**) for suffciently large **n** and **g**(**n**) is a coarse spectral bound of $\mathbf{G_n}$, then **f**(**n**) is also a coarse spectral bound of $\mathbf{G_n}$.

-Continuity: Suppose two graphs $\mathbf{G_n}$ and $\mathbf{G'_n}$ only differ by **o**(**dn**)-edges. If **f**(**n**) is a coarse spectral bound of **G**, then **f**(**n**) + **o**(**d**) is a coarse spectral bound of **G**'

Essential Spectral Radius

-The “least" coarse spectral bound is called essential spectral radius of **G**. Roughly speaking, an essential spectral radius is the absolute maximum of all but **o**(**n**) eigenvalues of **G**. By the monotonicity property, the essential spectral radius is well-defined up to a lower order additive term.

-It is known that the general random graph **G**(**n; P**) has essential spectral radius **O**(sqrt(n))

-By the continuity property and the Szemeredi's regularity lemma, one can show the essential spectral radius is **o**(**n**) for any family of graphs {$\mathbf{G_n}$}. In fact, we make the following conjecture.

-*Conjecture: For any family of graphs {G_n}, the essential spectral radius of G_n is O(sqrt(n))*

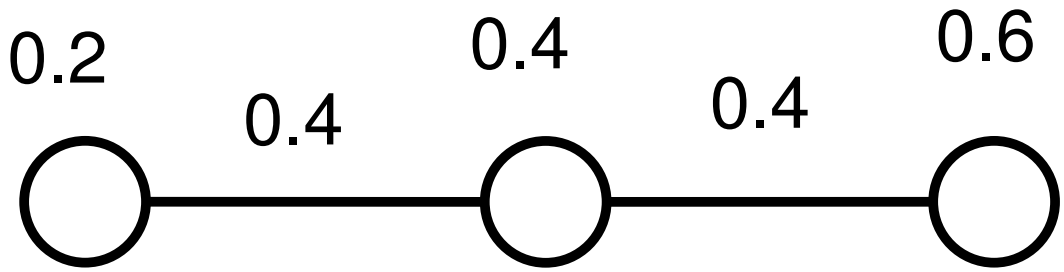
Distributed Algorithmic Blow-Up Lemma

- Need a way to algorithmically embed spanning subgraphs into dense host graphs.
- Need a distributed solution that (a) does not require global knowledge at each computational node, and (b) can potentially achieve a speedup from parallelism.
- The algorithmic version of Komlós, Sárközy, and Szemerédi's Blow-Up Lemma achieves this in the parallel computational model.
- We propose developing a distributed version of the algorithmic Blow-Up Lemma.

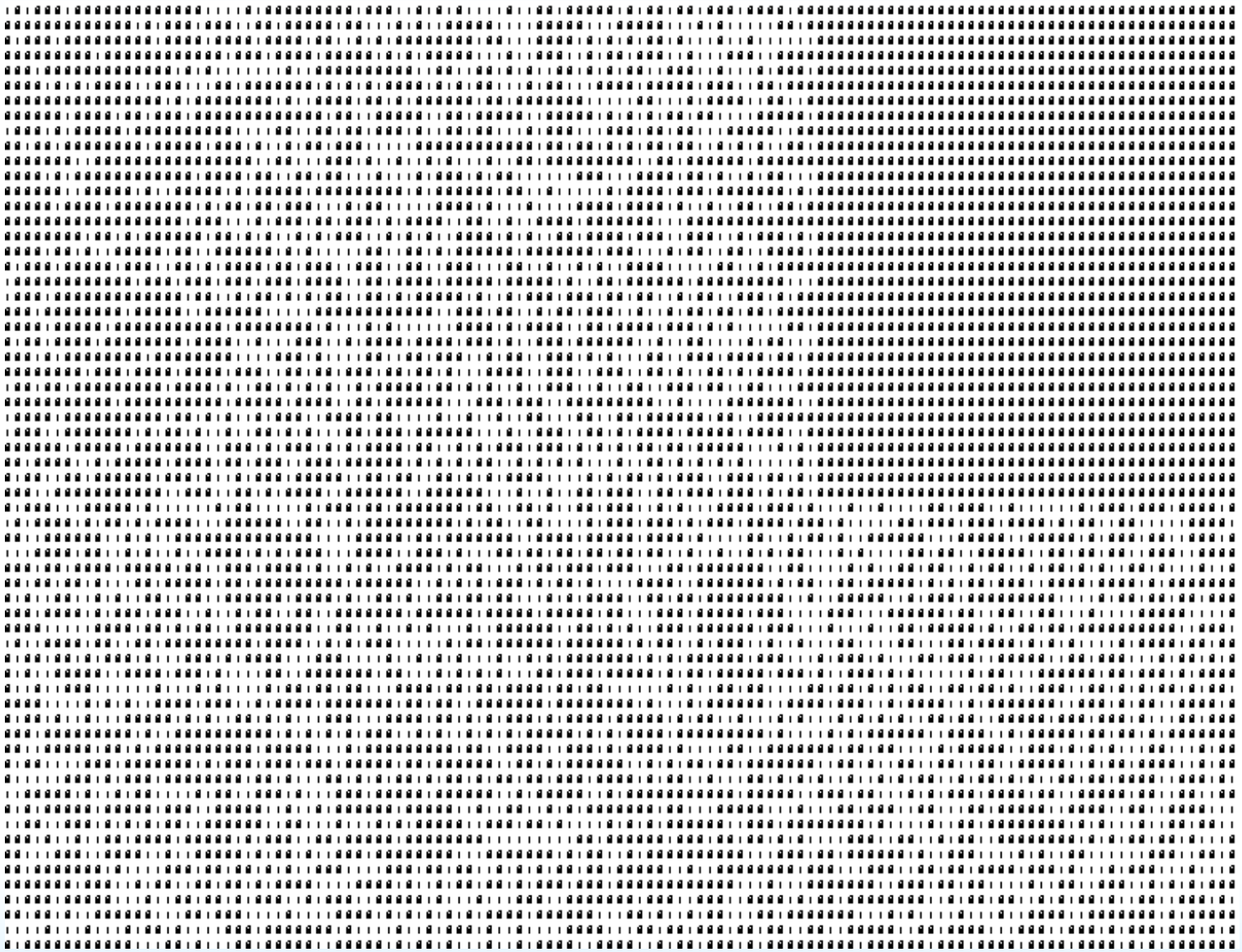
Distributed Algorithmic Blow-Up Lemma: Approach

- The algorithmic Blow-Up Lemma works in two phases:
  - phase 1 requires finding large independent sets; and
  - phase 2 requires solving an assignment problem.
- It was previously shown that phase 1 can be executed in the PRAM model in polylogarithmic time. *Our first contribution will be to show that the same result can be duplicated in a distributed computational model.*
- Phase 2 consists of finding a perfect matching between the remaining vertices and their potential embeddings. This can be cast as a Maximal Independent Set (MIS) problem. *Our next contribution will be a distributed algorithm for MIS.*

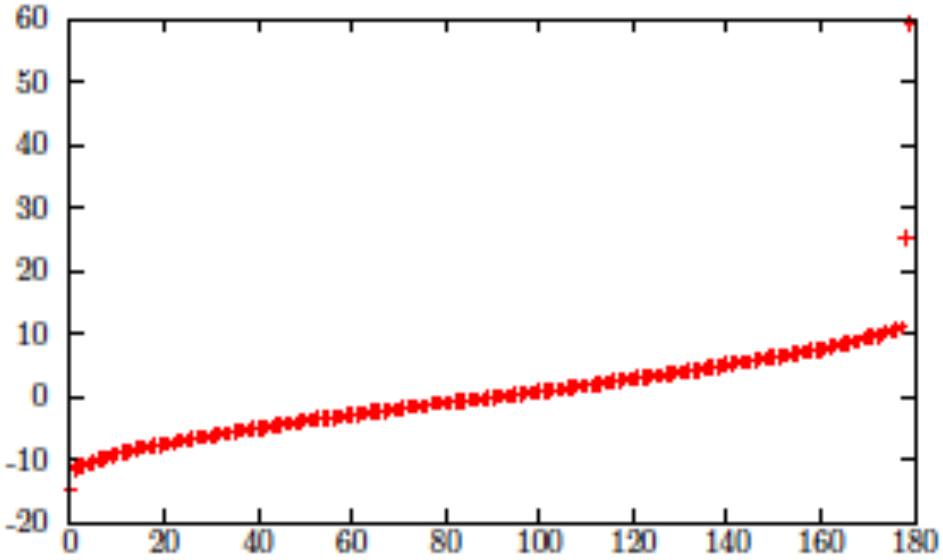
Example



Adjacency Matrix



Example



(a) Eigenvalues of G.

59.407

-25.300

14.809906

-11.457263

-10.892950

-10.759462

-10.296720

-9.846905

9.752137

-9.430681

-9.175593

-8.821417

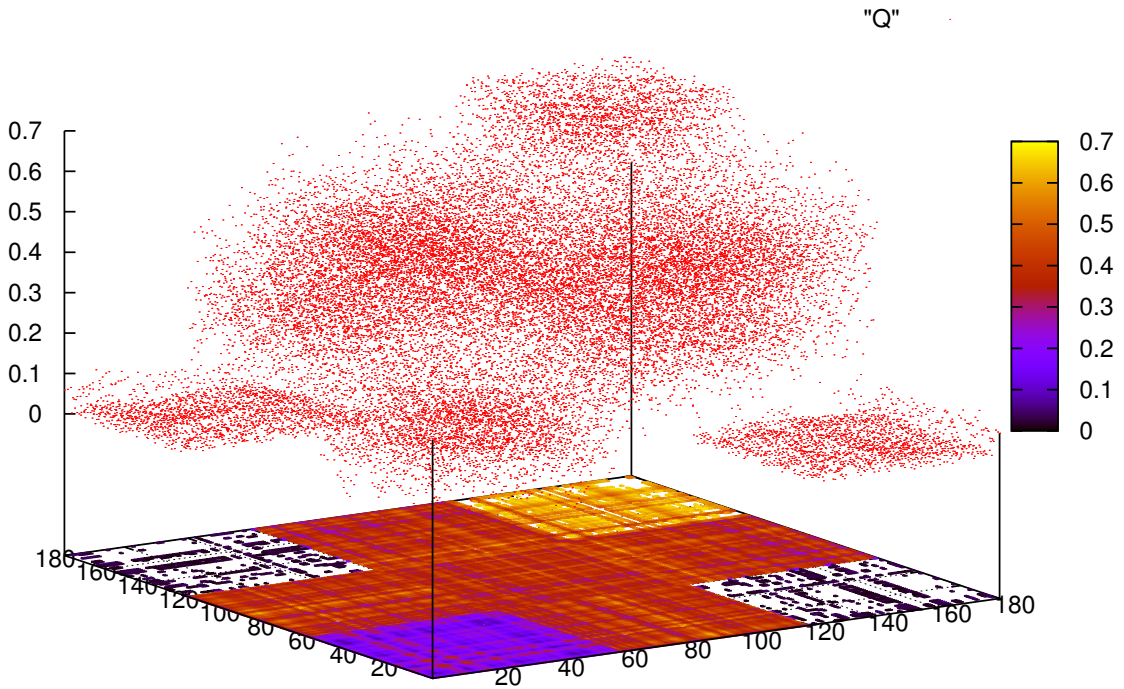
-8.634067

-8.573860

-8.411881

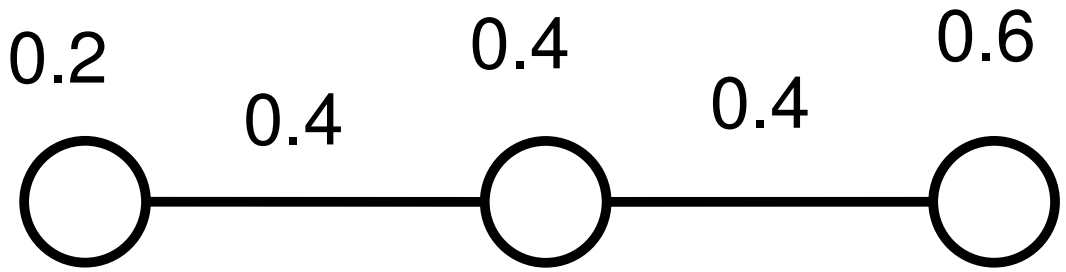
-8.198502

Example



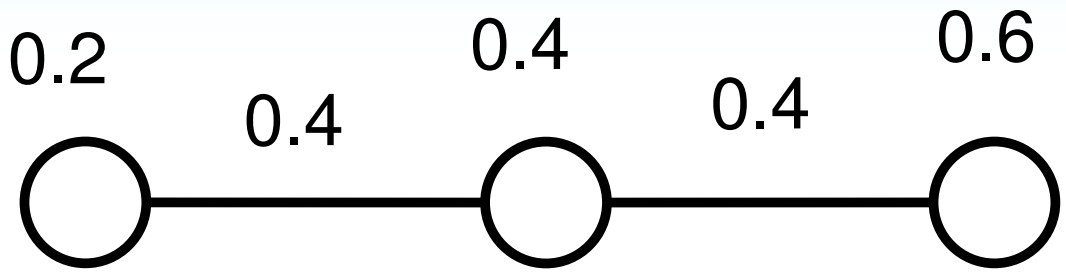
$$Q := \sum_{i=1}^{3} \lambda_i \alpha_i \alpha'_i.$$

P - Recovered



$$P = \begin{pmatrix} 0.198167 & 0.395086 & 0.002750 \\ 0.395086 & 0.383684 & 0.390717 \\ 0.002750 & 0.390717 & 0.595066 \end{pmatrix}.$$

Applications for number of triangles

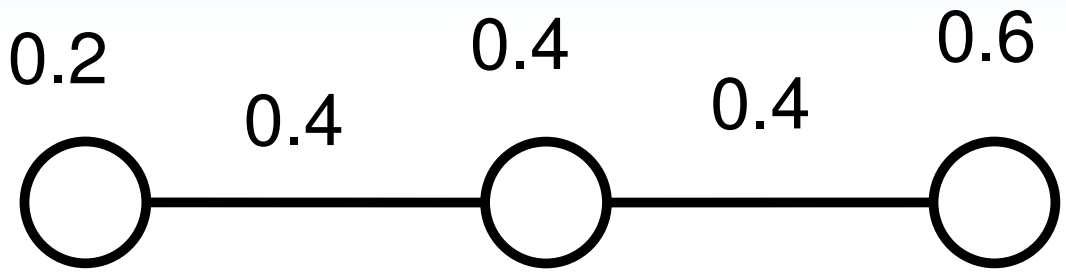


$$P = \begin{pmatrix} 0.198167 & 0.395086 & 0.002750 \\ 0.395086 & 0.383684 & 0.390717 \\ 0.002750 & 0.390717 & 0.595066 \end{pmatrix}.$$

$$\#\text{ of triangles } \approx \frac{1}{6}\sum_{i,j,k} |V_i||V_j||V_k| p_{ij}p_{jk}p_{ik}.$$

Estimated # of Triangles = 35937

Applications for number of triangles



$$P = \begin{pmatrix} 0.198167 & 0.395086 & 0.002750 \\ 0.395086 & 0.383684 & 0.390717 \\ 0.002750 & 0.390717 & 0.595066 \end{pmatrix}.$$

$$\#\text{ of triangles } \approx \frac{1}{6}\sum_{i,j,k} |V_i||V_j||V_k| p_{ij}p_{jk}p_{ik}.$$

Estimated # of Triangles = 35937
Real # of Triangles = 35058
Relative Error = 2.5%

This is an $O(m^3)$ algorithm. This is efficient since $m \ll n$.

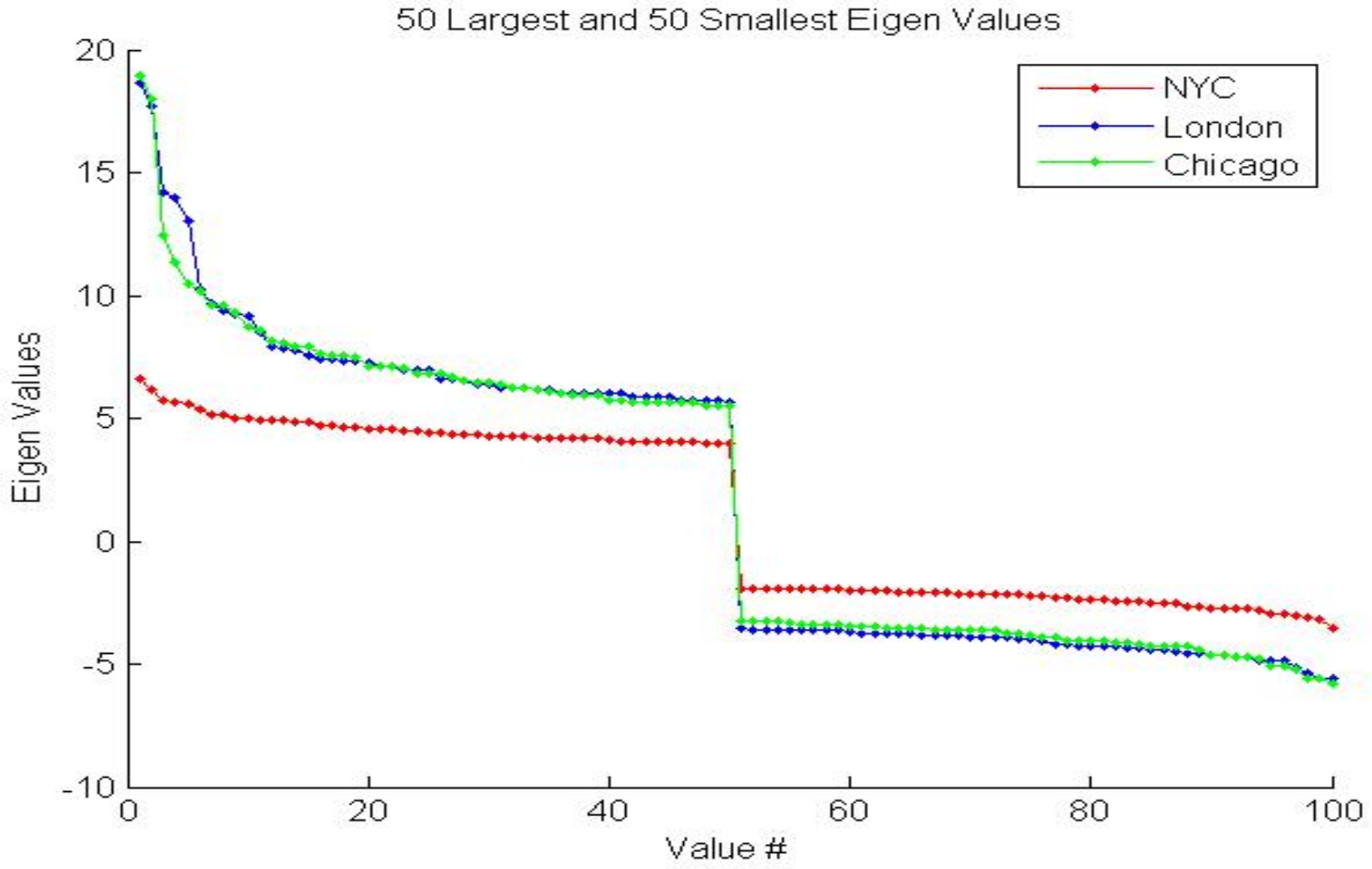
Real Twitter Data

Graph Name	Vertex #	Edge #	Driver Nodes	Persistent Driver Nodes (200 runs)
Chicago_graph	15096	38731	5170(34.2%)	3402(22.5%)
London_graph	27680	70685	10663(38.5%)	7686(27.7%)
Nyc_graph	8463	16365	2188(25.8%)	1529(18%)
Chicago_graph_1hop	155165	1522747	63803(41.1%)	14293(9.2%)
London_graph_1hop	185247	1491553	70622(38.1%)	18386(9.9%)
Newyork_graph_1hop	160237	1593929	75202(46.9%)	14247(8.9%)

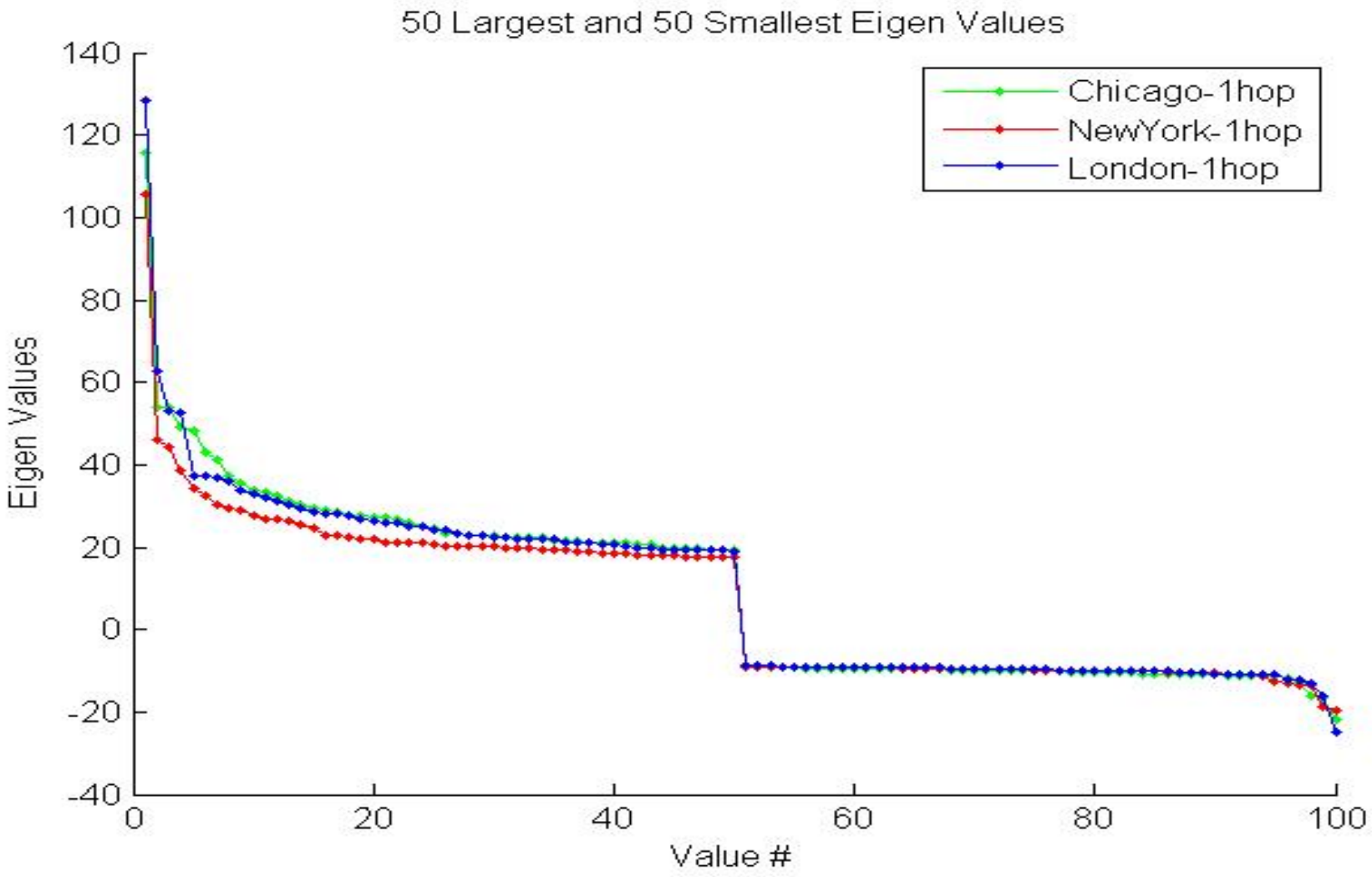
Adjacency Matrix Eigen Values

Graph Name	Largest	Second Largest	Smallest	Second Smallest
Chicago_graph	18.9	17.98	-5.8	-5.59
London_graph	18.61	17.67	-5.57	-5.57
Nyc_graph	6.56	6.18	-3.5	-3.16
Chicago_graph_1hop	115.85	54.1	-21.6	-16.39
London_graph_1hop	128.21	62.85	-25.0	-16.06
Newyork_graph_1hop	105.45	46.2	-19.79	-18.86

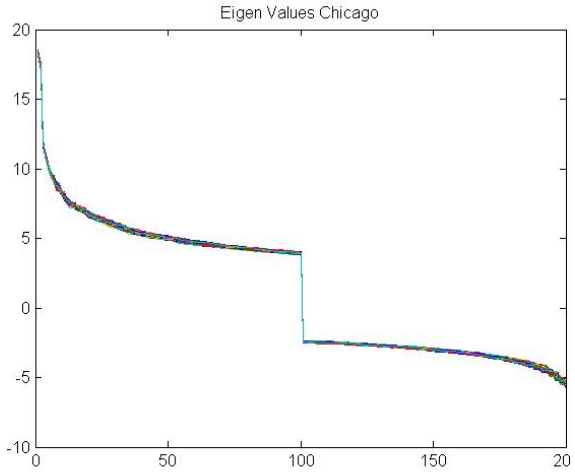
NYC is different from London and Chicago

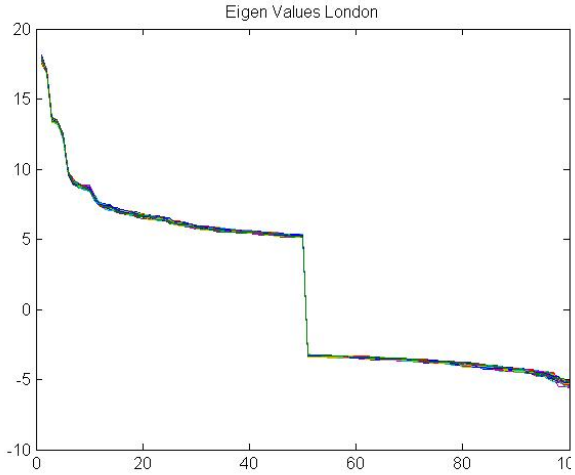


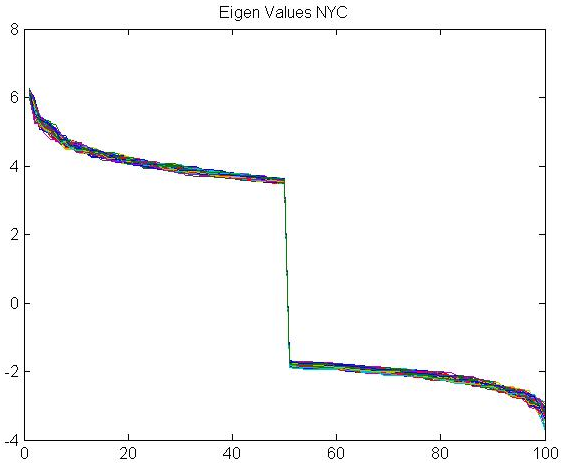
No Significant Difference at 1 hop out



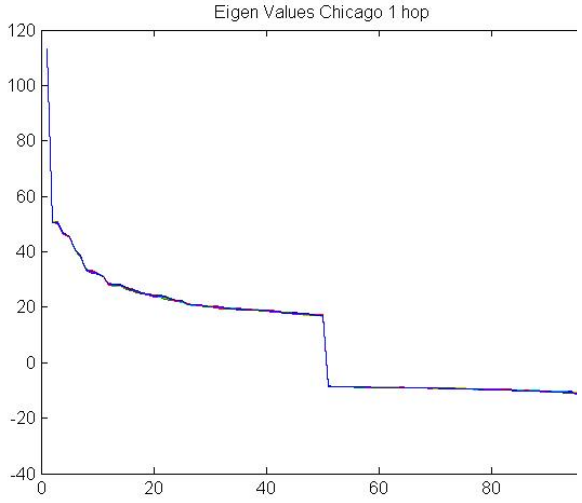
Randomization of Graphs and Eigen Value Trends

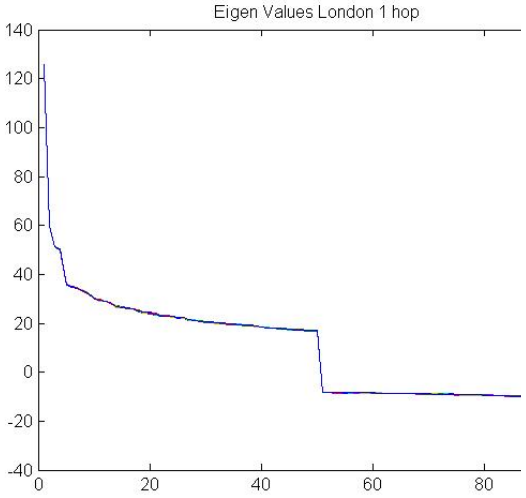


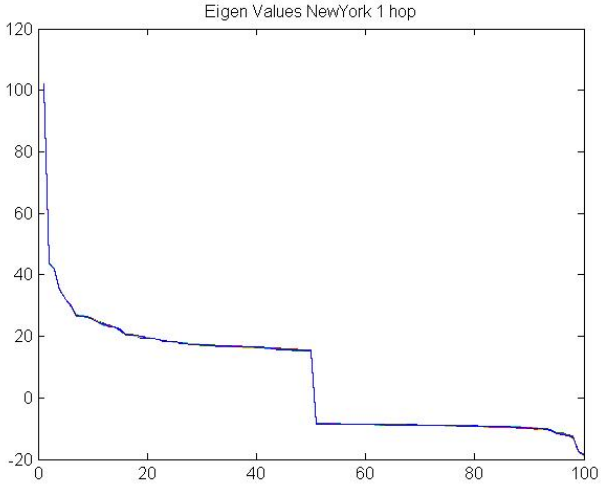




Randomization of Graphs and Eigen Value Trends continued







Why spectral method works?

- Many graphs can be approximated by general random graphs. For examples,
  - Dense graphs (by Szemeredi regularity Lemma),
- Parameters of general random graphs can be determined by the significant part of spectra.
- How about other graph models?
- Preferential Attachment Graphs

Definition of the Preferential Attachment Model

Let $m > 0$ be a fixed integer. Define a sequence of graphs $G_{m,1}, \ldots, G_{m,t}$ by the following recursive process:

- $G_{m,1}$ is a graph on one vertex, and m loops around this vertex.
- Given $G_{m,t-1}$ we construct $G_{m,t}$ by adding vertex t, and we connect it to m random points $Y_1, \ldots, Y_m$ which are iid with the following distribution:

$$\mathbf{P}[Y_i = u] = \frac{d(u, G_{m,t-1})}{2m(t-1)} \qquad \text{for } u \in [t-1]$$

We will refer to $G_{m,n}$ as a random Preferential Attachment graph (PA for short).

Graph Counts

Problem: Given a fixed graph H with a constant number of vertices, we want to know how many copies of H do we expect to see in a random PA graph $G_{m,n}$?

Example: Triangles in Erdös -Rényi

Let $G \sim G(n,p)$, and let H be a triangle. Denote by X_n to be the number of triangles in G. Then a trivial count yields:

$$\mathbf{E}[X_n] = \binom{n}{3} p^3$$

In general, if we have a small graph H with a vertices and b edges, then we would have: $\mathbf{E}[X_n] = \Theta\left(\binom{n}{a} p^b\right)$.

New results

Theorem

Let H be any admissible, ordered graph with a constant number of vertices and let $X_{m,n}$ count the number of copies of H in $G_{m,n}$. Denote by $f(H)$ the number of vertices of degree 1 in H, and denote by $g(H)$ the number of vertices of degree 2 in H. Then we have:

$$\mathbf{E}[X_{m,n}] = O\left(m^{|E(H)|} n^{f(H)/2} \log^{g(H)} n\right)$$

Moreover, one can replace $O(\cdot)$ by $\Theta(\cdot)$ if we have $d(v_i) \geq d(v_{i+1})$ where $\{v_1, v_2, \ldots, v_k\}$ are the vertices of H in order of appearance.

Graph Curvature

For a graph $G = (V, E)$, the *graph Laplacian* is the operator Δ on the space of functions $f : V \to \mathbb{R}$ given by

$$\Delta f(x) = \sum_{y \sim x} (f(y) - f(x)).$$

The *Bakry-Émery operators* are defined via

$$\Gamma(f, g) := \frac{1}{2} \left(\Delta(fg) - f\Delta g - g\Delta f \right)$$
$$\Gamma_2(f, g) := \frac{1}{2} \left(\Delta\Gamma(f, g) - \Gamma(f, \Delta g) - \Gamma(g, \Delta f) \right).$$

We write $\Gamma(f) := \Gamma(f, f)$ and $\Gamma_2(f) = \Gamma_2(f, f)$.

Graph Curvature

Definition (Bakry-Émery Curvature)

A graph G is said to satisfy the *curvature dimension inequality* $CD(K,n)$ for some $K \in \mathbb{R}$ and $n \in (0,\infty]$ if for all f,

$$\Gamma_2(f) \geq \frac{1}{n}(\Delta f)^2 + K \cdot \Gamma(f).$$

K is the curvature.
n is a “dimension."

Path Homology

Let $G = (V, E)$ be a directed graph.

Definition

An elementary path is a sequence $v_0 v_1 \cdots v_k$ of vertices. It is allowed if it corresponds to a directed walk in G.
A path is a formal linear combination of elementary paths.

Definition

The boundary operator ∂_n maps length n paths to paths of length $n-1$; for elementary paths it is defined as:

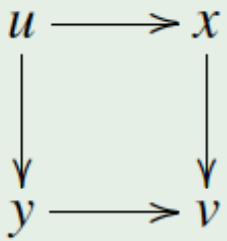
$$\partial(v_0 \cdots v_k) = \sum_{i=0}^{k} (-1)^i v_0 \ldots \widehat{v_i} \ldots v_k$$

where $\widehat{v_i}$ denotes omission of the ith entry.

Path Homology cont.

Clearly the image of an allowed path under ∂ might not be allowed. An allowed path p is ∂-invariant if ∂p is allowed too.

Example



$\partial(uxv) = xv - uv + ux$, so the allowed path uxv is not ∂-invariant.
But $uxv - uyv$ is ∂-invariant.

Path Homology cont.

Denote by Ω_n the set of all ∂-invariant paths of length n. These form the chain complex

$$\cdots \to \Omega_{n+1} \xrightarrow{\partial_{n+1}} \Omega_n \xrightarrow{\partial_n} \Omega_{n-1} \xrightarrow{\partial_{n-1}} \cdots \Omega_1 \xrightarrow{\partial_1} \Omega_0 \xrightarrow{\partial_0} 0.$$

Definition

$$H_n = \operatorname{Ker} \partial_n / \operatorname{Im} \partial_{n+1}$$

Questions

- If G satisfies $CD(0,\infty)$, is $dim\ H_1(G) < \infty$?
- Is there a nice combinatorial interpretation of H_k, $k > 1$?
- In a random graph (e.g. $G(n,p)$) what is the probability a vertex has positive curvature?
- In a random graph, what do we expect in terms of the homology? What is the probability that a random graph will have trivial H_1?